

# **CSE 259 - Logic in Computer Science (Spring 2024)**

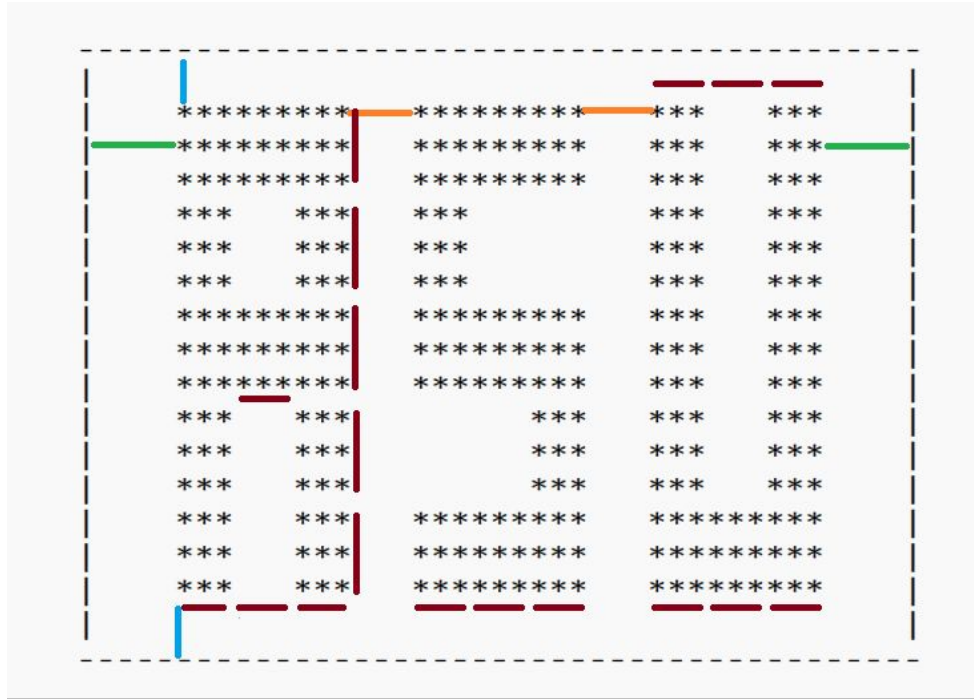
## **Recitation-4**

**Waqar Hassan Khan**



# Project-1

- Use the query ***asu(LeftRightMargin, BottomTopMargin, SpaceBetweenCharacters, FontSize)***
- The output should look like the following



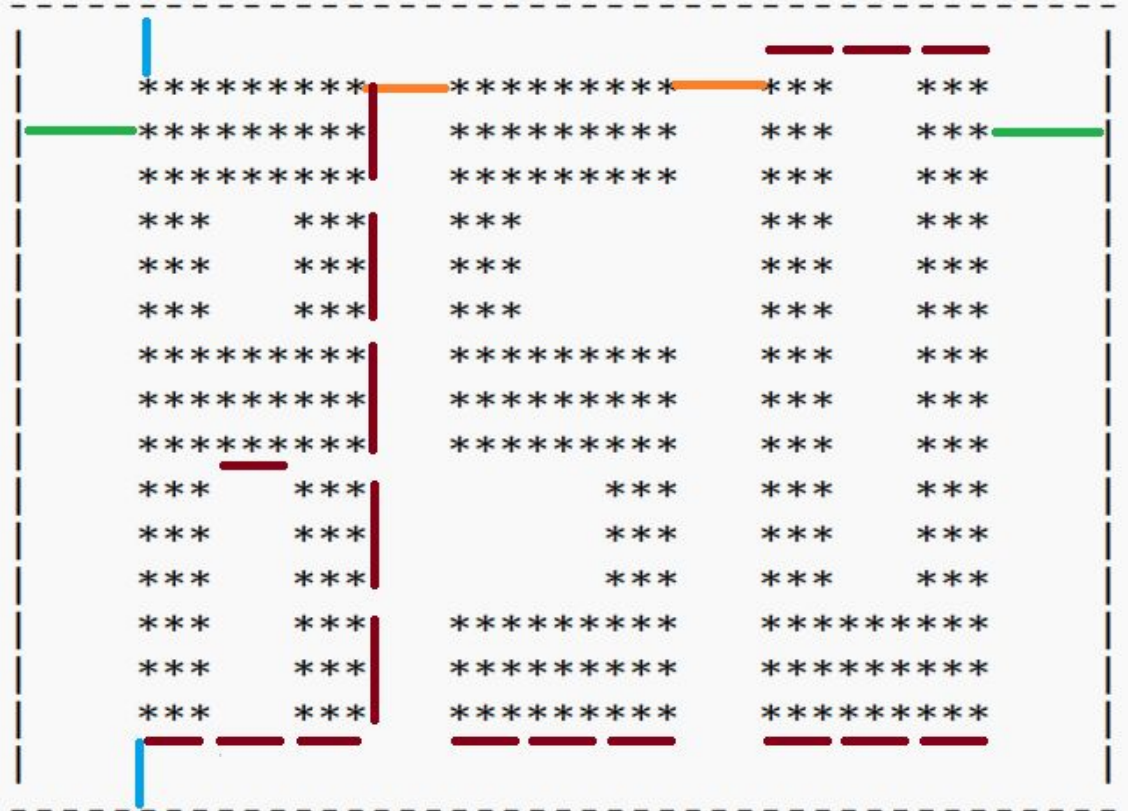
# Project-1

LeftRightMargin = 4

BottomTopMargin = 1

SpaceBetweenCharacters = 3

FontSize = 3



## Understanding the segment and FontSize

[illegible]

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*       \*\*\*  
\*\*\*       \*\*\*  
\*\*\*       \*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*       \*\*\*  
\*\*\*       \*\*\*  
\*\*\*       \*\*\*  
\*\*\*       \*\*\*  
\*\*\*       \*\*\*  
\*\*\*       \*\*\*

\*\*\* — \*\*\*  
— —

- Each letter can be divided horizontally into 3 segments.
- Each segment will have FontSize number of characters
- Each letter can be divided vertically into 5 segments
- Each segment will have FontSize number of characters

# Utility codes

```
drawSymbol(Symbol, 0).  
drawSymbol(Symbol, N) :- N > 0, write(Symbol), N1 is N - 1, drawSymbol(Symbol, N1).
```

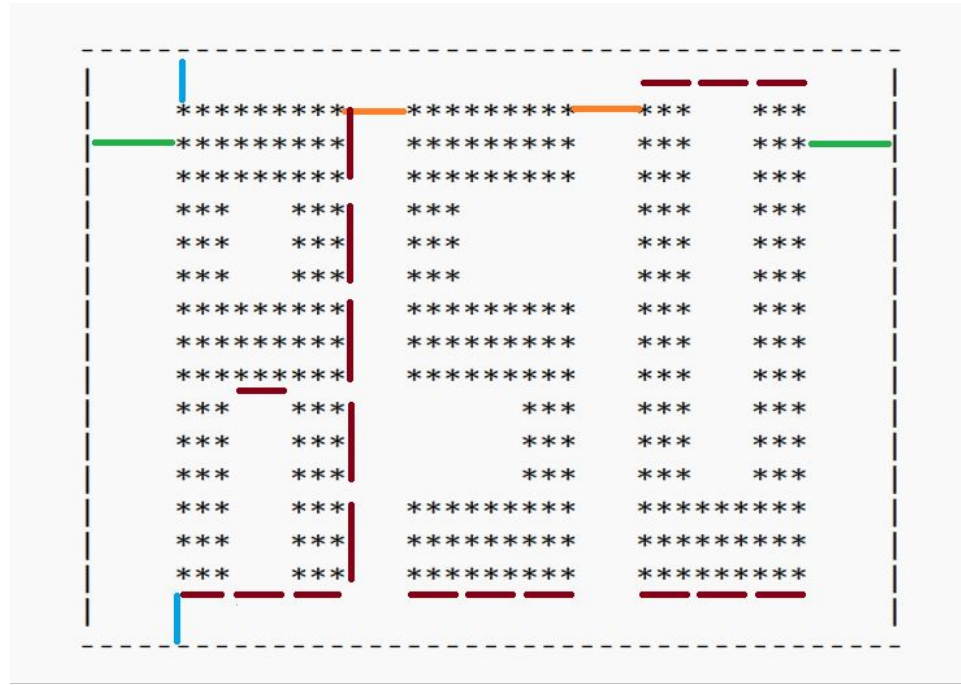
- If  $N = 0$  then, nothing to print
- Else,
  - we check if  $N > 0$
  - If true then print the character of our choice and move forward
  - We assign  $N1 = N - 1$
  - Call the rule recursively with the new value of  $N$

# Drawing the rectangle - horizontal line

Total characters = (LeftRightMargin x 2) + (SpaceBetweenCharacters x 2) + **(3 x 3 x FontSize) + 2**

**(3 x 3 x FontSize)**: 3 letters. Each character can be divided into 3 segments horizontally. Each segment has FontSize number of characters.

**+2 is for the 2 vertical bars!**



# Drawing the rectangle - horizontal line

```
drawSymbol(Symbol, 0).  
drawSymbol(Symbol, N) :- N > 0, write(Symbol), N1 is N - 1, drawSymbol(Symbol, N1).  
  
drawHorizontalLine(Symbol, 0) :- nl.  
drawHorizontalLine(Symbol, N) :- drawSymbol(Symbol, N).
```

```
| ?- drawHorizontalLine(-, 10).
```

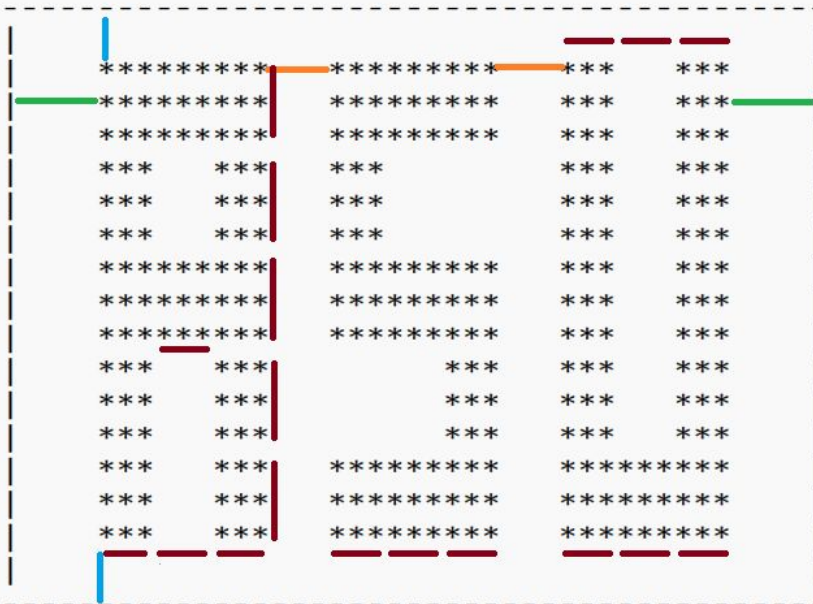
```
-----
```

```
true ?
```

```
yes
```

# Drawing the rectangle - vertical line

Total characters = (BottomTopMargin x 2) + (5 \* FontSize)





# Drawing the rectangle - vertical line

```
drawVerticalLines(Symbol, 0, Width).  
drawVerticalLines(Symbol, Height, Width) :-  
    Height > 0,  
    write(Symbol),  
    drawSymbol(' ', Width - 2),  
    write(Symbol),  
    nl,  
    Height1 is Height - 1,  
    drawVerticalLines(Symbol, Height1, Width).
```

# Drawing the rectangle - vertical line

```
| ?- drawVerticalLines('|', 10, 10).
```

```
|  
|  
|  
|  
|  
|  
|  
|  
|  
|  
|
```

```
true ?
```


```
yes
```

```
| ?-
```

# Drawing the rectangle - combining everything

```
drawRectangle(LeftRightMargin, BottomTopMargin, SpaceBetweenCharacters, FontSize) :-  
  integer(LeftRightMargin), integer(BottomTopMargin), integer(SpaceBetweenCharacters), integer(FontSize),  
  Width is (LeftRightMargin * 2 + SpaceBetweenCharacters * 2 + FontSize * 3 * 3 + 2),  
  Height is (BottomTopMargin * 2 + FontSize * 5),  
  drawHorizontalLine('-', Width),  
  nl,  
  drawVerticalLines('|', Height, Width),  
  drawHorizontalLine('-', Width).
```

# Drawing the rectangle - combining everything



The screenshot shows a window titled "GNU Prolog console" with a menu bar containing "File", "Edit", "Terminal", "Prolog", and "Help". The main text area contains the following text:

```
yes  
| ?- drawRectangle(4, 1, 3, 3).  
  
true ?
```

A dashed rectangle is drawn on the screen, representing the output of the `drawRectangle(4, 1, 3, 3).` command. The rectangle is positioned such that its top-left corner is at the start of the first line of the command, and its width and height correspond to the parameters 4 and 3 respectively.